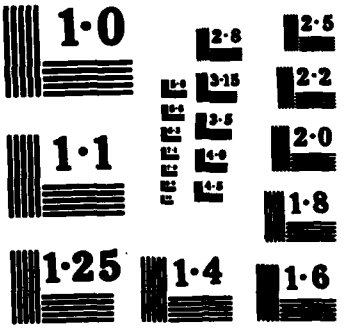
AD-A145 622 ON 'UPDATE SEMANTICS AND RELATIONAL VIEWS'(U) STANFORD 1/1
UNIV CA DEPT OF COMPUTER SCIENCE A M KELLER 1984
AFOSR-TR-84-0710 AFOSR-80-0212

UNCLASSIFIED F/G 9/2 NL


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0

(2)

AD-A145 622

REPORT DOCUMENTATION PAGE

1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S): AFOSR-TR. 84-0710
6a. NAME OF PERFORMING ORGANIZATION: Stanford University	6b. OFFICE SYMBOL (If applicable):
7a. NAME OF MONITORING ORGANIZATION: Air Force Office of Scientific Research	
6c. ADDRESS (City, State and ZIP Code): Department of Computer Science, Stanford CA 94305	7b. ADDRESS (City, State and ZIP Code): Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: AFOSR	8b. OFFICE SYMBOL (If applicable): NM
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: AFOSR-80-0212	
8c. ADDRESS (City, State and ZIP Code): Bolling AFB DC 20332	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2304	A7	

11. TITLE (Include Security Classification): "ON "UPDATE SEMANTICS AND RELATIONAL VIEWS""

12. PERSONAL AUTHOR(S): Arthur M. Keller

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Technical	FROM _____ TO _____	1984	2

16. SUPPLEMENTARY NOTATION: Supported in part by Contract N00039-82-G-0250 (the Knowledge Base Management Systems Project, Prof Gio Wiederhold, principal investigator, Defense Adv Rsch Projects Agency).

17. COSATI CODES		
FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Relational databases, database theory, complementary mappings, view update.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

A shared database encompasses data of interest to a variety of users. A database view provides a class of users with an image of a portion of the data presented according to the needs of these users. The ability to translate updates specified against the view into updates specified against the database is necessary to allow more effective use of views. Since a user accessing the database through a view has limited knowledge of the entire domain of the database, it is necessary to limit the effect on others of a particular user's view update. Furthermore, there may be many ways to translate a particular view update into database updates. Bancilhon and Spyratos propose the notion of a constant complementary view, which partially solves the problem of view updates by addressing these two issues. The authors present a reasonable view update translator that does not preserve any complement. This illustrates the overly restrictive consequences of the requirement that a complement remain constant.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
Dr. Robert N. Buchal	(202) 767- 4939	NM

DTIC ELECTE AUG 8 1984 S E

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE. UNCLASSIFIED SECURITY CLASSIFICATION OF THIS PAGE

On "Update Semantics and Relational Views"

Arthur M. Keller
Computer Science Dept., Stanford University

ABSTRACT. A shared database encompasses data of interest to a variety of users. A database view provides a class of users with an image of a portion of the data presented according to the needs of these users. The ability to translate updates specified against the view into updates specified against the database is necessary to allow more effective use of views. Since a user accessing the database through a view has limited knowledge of the entire domain of the database, it is necessary to limit the effect on others of a particular user's view update. Furthermore, there may be many ways to translate a particular view update into database updates. Bancilhon and Spyratos propose the notion of a constant complementary view, which partially solves the problem of view updates by addressing these two issues. We present a reasonable view update translator that does not preserve any complement. This illustrates the overly restrictive consequences of the requirement that a complement remain constant.

KEYWORDS. Relational databases, database theory, complementary mappings, view update.

CR CATEGORIES. H.2.1, H.1.1, E.4.

1. Introduction

We wish to control the effect of the actions of users of shared databases on other users without unnecessarily restricting these actions. Views provide an image of a portion of the database according to the user's needs [Stonebraker 75]. The problem of translating an update specified against the view into an update specified against the database has been explored [Bancilhon 81, Dayal 82, Keller 82] but not completely solved. One consideration is that various alternatives may exist, all of which implement the request desired by the user from the user's point of view. However, some of these translations may make unnecessary changes to others part of the database that do not affect the view.

Bancilhon and Spyratos [81] propose that a complementary view – one that contains all the information in the database not contained in the user's view – be held constant in order to preclude these "side effects" that may affect other users. This approach provides that any translation from a view update to a database update must be unique. Unfortunately, this rules out many reasonable translations that are otherwise acceptable. We present a particular view update translator that is quite reasonable, but that does not preserve any complement.

2. Definitions

We assume the reader is familiar with relational database theory as presented by Ullman [82] and Maier [83]. Prior work on complements [Bancilhon 81, Keller 84] will also provide useful background.

DEFINITION [Bancilhon 81]. Let f and g be two functions whose domain is D. Then f and g are *complementary mappings* if

$$[\forall x, y \in D][(x \neq y) \wedge f(x) = f(y) \rightarrow g(x) \neq g(y)].$$

COROLLARY. Given a database D and a view v and a complementary view c, there is at most one database state that corresponds to a desired view state (range of v) for a fixed view state (range of c).

The consequence of this corollary is that a view update translator that holds a complement constant has at most one translation. There are, however, view update translators that have at most one translation that do not hold any complement constant. In the next section, we will a reasonable one.

3. A View Update Translator

Consider the relation AB, with two attributes A and B, and the functional dependency $A \rightarrow B$. Let the domain of A contain at least one element, a1, and the domain of B contain at least two elements, b1 and b2. We define the view V to select all tuples from AB where B = b1.

We shall define a view update translator that accepts all single tuple updates valid in the view.

This work was supported in part by contract N00039-82-G-0250 (the Knowledge Base Management Systems Project, Prof. Gio Wiederhold, Principal Investigator) from the Defense Advanced Research Projects Agency and by contract AFOSR-80-0212 (Universal Relations, Prof. Jeff Ullman, Principal Investigator) from the Air Force Office of Scientific Research, both of the United States Department of Defense. The views and conclusions contained in this document are those of the authors and should not be interpreted as representative of the official policies of DARPA or the US Government.

Author's address: Computer Science Department, Stanford University, Stanford, CA 94305-2085.

Insert tuple ⟨a, b⟩: If there exists a tuple ⟨a, y⟩, then replace ⟨a, y⟩ with ⟨a, b⟩, otherwise insert ⟨a, b⟩.

Delete tuple ⟨a, b⟩: Delete tuple ⟨a, b⟩ from the underlying database.

Replace tuple ⟨a, b⟩ by tuple ⟨c, d⟩: Perform translation for deleting ⟨a, b⟩ followed by translation for inserting ⟨c, d⟩.

Let us consider the translations of the insertion of the tuple ⟨a1, b1⟩ starting with two different database states using this view update translator.

Initial database state 1:
A *B*
a1 b2

Initial view state 1:
A *B*
(empty relation)

Result database state 1:
A *B*
a1 b1

Result view state 1:
A *B*
a1 b1

Initial database state 2:
A *B*
(empty relation)

Initial view state 2:
A *B*
(empty relation)

Result database state 2:
A *B*
a1 b1

Result view state 2:
A *B*
a1 b1

We observe that initial view state 1 and initial view state 2 are the same, yet initial database state 1 and initial database state 2 are different. Therefore, any complement view must have different values for initial database state 1 and initial database state 2. However, the result database states are the same. Thus, the result complement states must be the same. Consequently, the complement cannot be remain constant.

If we wanted to hold constant a complement, we could, for example, choose the complement formed by selecting all tuples with $B \neq$ b1. This would preclude accepting the insertion request above for database state 1. We could define another translator that holds another complement constant, but it could not implement all of these requests in the same way.

4. Conclusion

While view complements provide insight into the process of view update translation, requiring that a complement be chosen that remains constant is too restrictive. Bancilhon and Spyratos [81] prove that alternative (minimal) complements exist, but do not indicate how to generate all of them. They also do not show how to derive a view update translator given a constant complement. We suggest that further work consider the generation of alternative view update translations with limited effects on parts of the database not appearing in the view.

Bibliography

[Bancilhon 81] F. Bancilhon and N. Spyratos, "Update Semantics and Relational Views," *ACM Trans. on Database Systems*, 6:4, December 1981.

[Dayal 82] U. Dayal and P. A. Bernstein, "On the Correct Translation of Update Operations on Relational Views," *ACM Trans. on Database Systems*, 7:3, September 1982.

[Keller 82] Arthur M. Keller, "Updates to Relational Databases Through Views Involving Joins," in *Improving Database Usability and Responsiveness*, Peter Scheuermann, ed., Academic Press, New York, 1982.

[Keller 84] Arthur M. Keller and Jeffrey D. Ullman, "On Complementary and Independent Mappings on Databases," *1984 ACM/SIGMOD Int. Conf. on Management of Data*, Boston, June 1984.

[Maier 83] D. Maier, *Theory of Relational Databases*, Computer Science Press, Rockville, MD, 1983.

[Stonebraker 75] Michael Stonebraker, "Implementation of Integrity Constraints and Views by Query Modification," *Proc. of the 1975 SIGMOD Conference*, ACM SIGMOD, San Jose, June 1975.

[Ullman 82] Jeffrey D. Ullman, *Principles of Database Systems*, Computer Science Press, Potomac, MD, second edition, 1982.

END

FILMED

10-84

DTIC